

\ Connecting jotblk /
and uEforth
/ \

May 10, 2025

Brad Nelson

73

Motivation

- * I use a bunch of different laptops
- * I often have Forth related ideas
- * Too many options overwhelm me
- * I like blocks

74

Blocks on the Web

- * I've done this before, RainbowForth 2008
- * And with uEforth on the Web
-- AND Forgot!
- * The first Forth I really used
was Pygmy Forth,
and the blocks stuck...

75

Implementation

- * AppEngine
 - Google's Cloud Turnkey option
 - DataStore API (Now Firebase)
 - Mostly just a Python venv now
- * Flask
 - Micro Web framework
- * JavaScript
 - Render to a canvas
 - Stretch monospace font to fit 64x16+1

76

Block Storage

- * Types of Requests:
 - READ start $\leq i < \text{end}$ blocks
 - WRITE a block
- * Keep 16,000 Blocks in memory
- * Keep flags: DIRTY, LOADED
- * Load blocks when first seen
 - Chunks of 64 blocks
- * Save blocks when they change
 - Check every 3 seconds

77

For Whom?

- * Blown lots of worry about making blocks work for the world
- * Focus on me as the customer
- * Authenticate somehow...
- * Worry about generalizing later
- * Just for me, for now...

78

Editor

- * Arrows to move, PgUp/PgDn move screens
- * QWERTY to type, Home/End
- * Insert/Delete to 2 spaces
- * No shadow blocks
- * 64x16+1 status line, fills window
- * Autosave after 3 seconds
- * ^O - Find empty (open) block
- * ^L - Create log entry
- * ^Enter - Follow @1 / (search) links
- * ^B - go back

What about Forth?

- * Was tempted to do another...
- * But uEforth is flexible
- *
 - Interface to add words
 - Control bootup
 - Activate via ^F
 - Shared read-only BLOCK storage

Forth Injection

```
JSWORD: block-read { a n -- }  
  for (var i = 0; i < 1024; i++) {  
    u8[a + i] = blocks[n * 1024 + i];  
  }  
~
```

```
variable scr  
create block-buffer 1024 allot  
: block ( n -- a ) block-buffer over block-read ;
```

81

DEMO

QUESTIONS?

Thank you!

82